

000 DYNAMIC INFILLING ANCHORS FOR FORMAT- 001 002 CONSTRAINED GENERATION IN DIFFUSION LLMs 003 004

005 **Anonymous authors**

006 Paper under double-blind review

007 008 ABSTRACT 009

011 Diffusion large language models (dLLMs) have recently emerged as a compelling
012 alternative to autoregressive LLMs, offering bidirectional attention and parallel
013 sequence generation. These properties allow dLLMs to exploit global contextual
014 information and naturally support the integration of non-sequential constraints,
015 making them particularly suitable for format-constrained tasks such as generat-
016 ing parseable JSON or reasoning–answer templates. A straightforward approach
017 is to enforce such constraints with fixed anchors, but this often results in rigid
018 generation spans, leading to truncated reasoning or redundant content. To over-
019 come this limitation, we propose a training-free method, Dynamic Infilling An-
020 chors (DIA). DIA dynamically adjusts generation length by estimating appro-
021 priate end-anchor positions before content generation, followed by iterative in-
022 filling between anchors. This flexible mechanism ensures structural correctness
023 and semantic coherence while avoiding the inefficiencies of fixed-span methods.
024 Experiments on reasoning-oriented benchmarks demonstrate that DIA substan-
025 tially improves both format compliance and answer accuracy, achieving signif-
026 icant gains on GSM8K and MATH under zero-shot settings. These results high-
027 light the promise of dLLMs for reliable, structure-aware generation and establish
028 DIA as a practical pathway toward robust format-constrained text generation.

029 1 INTRODUCTION 030

031 In recent years, diffusion large language models (dLLMs)(Nie et al., 2025; Ye et al., 2025; Labs
032 et al., 2025; Song et al., 2025; Deepmind, 2024) have attracted increasing attention due to their
033 distinctive computational mechanisms and promising potential. Unlike traditional autoregressive
034 language models (AR LLMs), which rely on left-to-right sequential decoding, dLLMs are not re-
035 stricted to unidirectional dependencies during generation. Instead, they employ a bidirectional at-
036 tention mechanism, enabling the model to update token representations at each step by leveraging
037 complete contextual information simultaneously. This mechanism allows all positions in a sequence
038 to be predicted in parallel rather than generated step by step, thereby substantially enhancing both
039 modeling flexibility and computational efficiency. Beyond efficiency gains, this parallelism also
040 strengthens the contextual modeling capacity of dLLMs, enabling them to capture global depen-
041 dencies more comprehensively.

042 Within this property, we identify not only the potential to enhance contextual modeling and gen-
043 eration efficiency, but also the possibility of directly incorporating non-sequential constraints into
044 the generation process. The exposure of a fully masked sequence in dLLMs allows us to impose
045 global constraints on the target output by directly editing the masked sequence. For instance, one
046 may preemptively replace selected mask tokens with predetermined conclusions or mandatory con-
047 tent, thereby guiding the model toward iterative optimization under the specified requirements. This
048 observation motivates us to explore the application of dLLMs to the problem of format-constrained
049 generation. The term refers to scenarios in which the model’s output must strictly adhere to pre-
050 defined structures and requirements. For example, producing parseable JSON representations. To
051 evaluate this capability, we adopt a representative thinking–answering task as the testing scenario,
052 where existing dLLMs fail to achieve satisfactory outcomes.

053 To address these challenges, a straightforward approach is to enforce structural constraints by insert-
054 ing anchors (e.g. $\langle \text{think} \rangle$, $\langle / \text{think} \rangle$, $\langle \text{answer} \rangle$, $\langle / \text{answer} \rangle$) directly into the masked

054 sequence. However, while this approach appears intuitive, it also introduces new challenges. Once
 055 anchor positions are fixed in advance, the generative space between them becomes rigid, forcing
 056 the model to allocate tokens within predetermined boundaries. Such rigidity can lead to suboptimal
 057 allocation of generative space and ultimately impair output quality. In practice, when the fixed span
 058 between anchors is too short, the reasoning process is often truncated before completion. On the
 059 other hand, when the span is too long, the model tends to produce redundant or repetitive content,
 060 thereby reducing both efficiency and reliability.

061 To obtain an appropriate generation length between anchors, thereby ensuring format correctness
 062 while maintaining generation quality, we propose a more flexible training-free strategy termed *Dy-
 063 namic Infilling Anchors (DIA)*. Our approach is inspired by previous studies on dLLMs(Li et al.,
 064 2025), which demonstrates that the model can estimate the position of the end token with only one
 065 or a few prediction steps, thereby determining a suitable generation length. We extend this capa-
 066 bility to predict the proper positions of anchors before content generation. Specifically, our method
 067 consists of two stages: (1) generation length adjustment by estimating position of the end anchor,
 068 and (2) iterative generation between fixed anchors.

069 The first stage of our method involves adjusting the generation space by estimating the position of
 070 the end anchor. Following the user prompt, the model initializes a relatively short, fully masked
 071 sequence, which serves as a starting point for the task output length and is dynamically extended
 072 later. For a think-answer task, this masked sequence is evenly divided into two blocks, with the
 073 corresponding begin anchors inserted at the start of each block. We then determine the anchor
 074 positions sequentially, one block at a time. Within each block, the model performs a single prediction
 075 step on the sequence, which is pre-filled with the begin anchor. If the prediction fails to produce an
 076 end anchor or yields one with insufficient confidence, it suggests that the current generation length
 077 is inadequate. Therefore, we extend the block by appending additional masked tokens to ensure
 078 adequate space for content generation and repeat the prediction step. This extension continues until
 079 the model successfully produces a valid end anchor or the block length reaches its upper limit.
 080 The design of Stage I fully leverages the model’s awareness of the generation space; it guarantees
 081 sufficient allocation for each phase while minimizing redundant space and unnecessary computation.

082 The second stage performs iterative generation after anchors are fixed. In the previous stage, we
 083 obtained a reasonable generation length and fixed the position of the end anchor. Based on this setup,
 084 we now generate the intermediate content between the anchors. This step effectively compensates
 085 for the limitations of single-step prediction and helps the model establish clear semantic boundaries
 086 across different segments, thereby promoting coherent content generation.

087 We validate the effectiveness of DIA on reasoning-oriented benchmarks. Experimental results on
 088 GSM8K(Cobbe et al., 2021) (0-shot) and MATH(Hendrycks et al., 2021) (0-shot) show that our
 089 method improves format correctness from 58.83% and 29.10% to **72.63%** and **76.82%**, respectively.
 090 Moreover, by better controlling the generation space, our method also improves answer accuracy
 091 from 14.86% and 19.52% to **46.78%** and **20.08%**, respectively. These results demonstrate that DIA
 092 substantially enhances both the reliability and quality of format-constrained generation with dLLMs.
 093 In summary, our contributions are three-fold:

- 094 1. We introduce a novel dLLM-based strategy for format-constrained generation.
- 095 2. We design a dynamic adjustment mechanism that flexibly allocates generative space, miti-
 096 gating the rigidity of fixed-anchor methods.
- 097 3. We will release code and resources to foster reproducibility and further research in this
 098 emerging area.

100 2 RELATED WORKS

101 **Diffusion Large Language Models** The evolution of diffusion paradigms in language modeling
 102 can be traced back to masked language models(Devlin et al., 2019), which randomly mask a subset
 103 of tokens in the input and predict the missing content, laying the groundwork for denoising-based
 104 generation. Building on this idea, early studies introduced continuous-space diffusion language
 105 models(Jo & Hwang, 2025), mapping text into continuous latent representations and generating se-
 106 quences through diffusion and reverse denoising. However, such methods suffered from ambiguity
 107 in representation and instability in decoding discrete text. To address this limitation, discrete-space

108 diffusion language models(Austin et al., 2023) were proposed, directly modeling diffusion and de-
 109 noising at the token level, thereby aligning the process more naturally with the discrete nature of lan-
 110 guage. Along this trajectory, BlockDiffusion(Arriola et al., 2025) incorporated block-wise modeling
 111 to mitigate the computational inefficiencies of diffusion-based text generation. For large-scale pre-
 112 training, a practical strategy for diffusion large language models (dLLMs) is to initialize them from
 113 pretrained autoregressive models(Gong et al., 2025a; Ye et al., 2025) and further align them with
 114 instructions to enhance task adaptability(Yang et al., 2025b; You et al., 2025; Song et al., 2025). To
 115 strengthen advanced capabilities, researchers have also explored reinforcement learning(Wang et al.,
 116 2025; Zhao et al., 2025; Gong et al., 2025b) as a post-training method. Meanwhile, dLLMs are being
 117 extended to multimodal scenarios through cross-modal alignment, enabling broader applications in
 118 understanding and generating modalities such as images and speech.
 119

120 **Format-Constraints** Format-constrained generation is critical for deploying language models, as
 121 it directly affects the parseability and reliability of code generation, structured outputs, and rea-
 122 soning templates. Existing studies often constrain the input side (prompt design(Ye et al., 2024)
 123 and example-based guidance(Min et al., 2022)), yet they are unstable under long-chain or high-
 124 complexity reasoning; output-side repair (post-processing and re-ranking(Gao et al., 2025; Zhuang
 125 et al., 2025)) improves format compliance but struggles to preserve semantic and structural consis-
 126 tency simultaneously. Fine-tuning or reinforcement learning on task-specific data(Song et al., 2025;
 127 Xiong et al., 2023; Cui et al., 2024; Yang et al., 2023) can enhance robustness, but the approach
 128 is costly and generalizes poorly across tasks. Constrained decoding(Mündler et al., 2025; Banerjee
 129 et al., 2025) with grammars or finite-state machines enforces strict compliance at the expense of
 130 efficiency and flexibility.
 131

132 **Large Language Models** The evolution of LLMs(Yang et al., 2025a; Grattafiori et al., 2024;
 133 DeepSeek-AI et al., 2025; Anthropic, 2025; Deepmind, 2025; xAI, 2025; OpenAI, 2025) has been
 134 fundamentally driven by insights from scaling laws (Kaplan et al., 2020), which reveal power-law
 135 relationships among model size, data, and compute, thereby guiding systematic capability improve-
 136 ments. Building on this foundation, researchers have observed the emergent phenomenon of in-
 137 context learning (ICL)(Min et al., 2022), whereby LLMs can rapidly adapt to new tasks from
 138 demonstrations without explicit parameter updates, showcasing remarkable transfer and general-
 139 ization abilities. To further enhance practical usability and alignment with human preferences, post-
 140 training techniques such as fine-tuning(Ouyang et al., 2022) and reinforcement learning(Schulman
 141 et al., 2017; Rafailov et al., 2024; Shao et al., 2024) have been extensively applied, playing a cen-
 142 tral role in task adaptation and alignment. In parallel, the rise of multimodal models has spurred
 143 advances in cross-modal alignment(Li et al., 2023; Liu et al., 2023), enabling LLMs to operate ef-
 144 fectively across text, vision, and speech, and thereby extending their versatility. Collectively, these
 145 lines of research have driven sustained progress in LLM capability, alignment, and applicability.
 146

147 3 METHOD

148 3.1 PRELIMINARY

149 **Inference of dLLMs.** In the generation stage of a diffusion language model (dLLM), the response
 150 sequence to be refined is initialized by concatenating the input prompt with a fully masked sequence
 151 of a specified length:
 152

$$153 x_t = \text{Concat}(\text{prompt}, \{[MASK]\}_{0: max_len-1}), \quad (1)$$

154 where max_len denotes the fixed response length. The generation process follows a discrete-time
 155 masked diffusion procedure, which can be formulated as a Markov chain. Thus, each prediction step
 156 depends only on the previous state, and in every iteration only the masked positions are updated in
 157 parallel:
 158

$$160 P_{0|t} = \prod_{s=t}^0 \prod_{i=0}^{n-1} P_{s|s+1}(x_s^i | x_{s+1}), \quad (2)$$

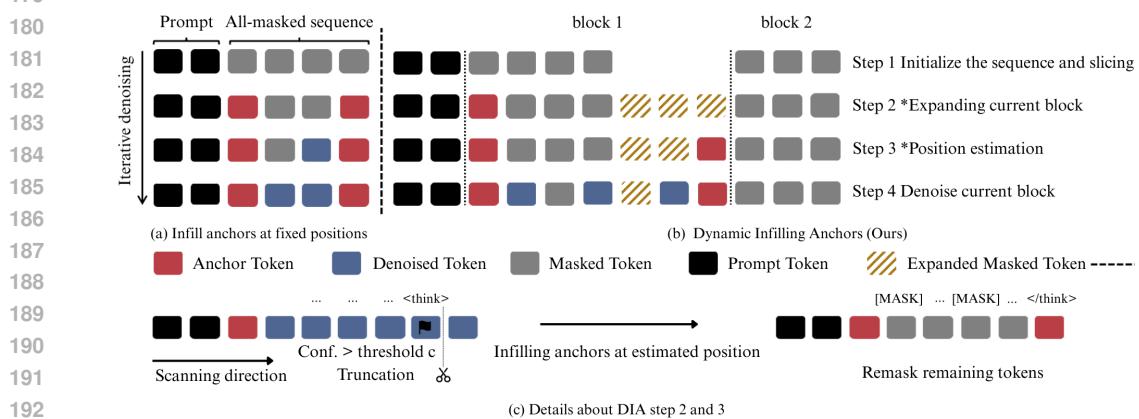
162

$$P_{s|s+1}(x_s^i | x_{s+1}) = \begin{cases} 1 & \text{if } x_{s+1}^i \neq [M], \text{ then } x_s^i = x_{s+1}^i, \\ 1 - \max(q(x_s^i)) & \text{if } (x_{s+1}^i = [M], \max(q(x_s^i)) < C), \\ \max(q(x_s^i)) & \text{if } (x_{s+1}^i = [M], \max(q(x_s^i)) \geq C) \\ & \text{or } s = 0, \text{ then } x_s^i \neq [M]. \end{cases} \quad (3)$$

170 Where $[M]$ denotes the $[MASK]$ token, $q(x_s^i)$ represents the output logits at position i in step s ,
 171 and C is the minimum confidence threshold.

173 3.2 DYNAMIC INFILLING ANCHOR

175 To overcome the limited flexibility of straightforward infilling methods in diffusion language
 176 models, we propose DIA, a training-free, two-stage approach. DIA selects an appropriate end-anchor
 177 position through a single-step prediction, thereby ensuring both format correctness and generation
 178 quality. The overview of our method is illustrated in Figure 1.



193 Figure 1: Dynamic Infilling Anchors (DIA). (a) Fixed-position infilling baseline. (b) Overview
 194 about our methods: DIA. (c) Details of expansion and anchor infilling steps with truncation and
 195 remasking.

197

198 3.2.1 GENERATION LENGTH ADJUSTMENT BY ESTIMATING POSITION OF THE END ANCHOR

200 DLLMs implicitly acquire a prior distribution over response termination positions from large-scale
 201 training corpora (Li et al., 2025). Specifically, for different input queries, the confidence of
 202 predicting the `eos` token at various positions within the answer sequence is not uniform, but instead
 203 exhibits a trend correlated with the appropriate response length. Building on this insight, we extend
 204 this capability to format-constrained tasks. For a typical reasoning-answer task, when the model
 205 receives the start anchor of a reasoning or answering section, it should be able to anticipate at what
 206 sequence length a corresponding “end-of-reasoning” or “end-of-answering” anchor is likely to
 207 occur. Intuitively, if the allocated generation space is sufficient to accommodate the reasoning or
 208 answering process, the one-step prediction will contain an end anchor (or partial end anchor) with
 209 high confidence exceeding a given threshold. Conversely, if the generation space is insufficient,
 210 the corresponding anchor will either fail to appear or appear only with substantially reduced confidence.

210

Building on this assumption, we design the generation-space estimation procedure of DIA. Given
 211 an input sequence X , which consists of the user query Q and a fully masked sequence X_L of a
 212 specified length L , DIA divides the sequence into two blocks ($\mathcal{C} = \{C_1, C_2\}$) of equal size (in terms
 213 of masked tokens), corresponding to the reasoning and answering stages. For each block, DIA first
 214 pre-fills the start anchor at the beginning of the decodable region. After inserting the start anchor, the
 215 block undergoes a one-step prediction. The prediction results and their associated confidence scores
 are used to determine whether the allocated generation length is appropriate. Since the model is

unlikely to produce a complete anchor token sequence in a single prediction, partial anchors are also incorporated into the decision mechanism. If the prediction either fails to produce an end anchor (or a partial end anchor) or yields an end anchor with confidence below the threshold c , the length of current block is expanded by a fixed length Δ , and the 'predict–decide' cycle is repeated until the generation space is sufficient to support the model in completing the reasoning or answering process. When multiple positions in the sequence satisfy the confidence threshold simultaneously, we retain the position closest to the left boundary to prevent the generation of duplicate end anchors within the sequence. To avert unbounded expansion, a maximum block length M is imposed. We truncate the redundant tokens following the selected end-anchor position and subsequently complete the partial end anchor to form a full one.

3.2.2 ITERATIVE DENOISING WITH INFILLING

In Stage I, we establish the block boundaries by determining the positions of the anchors. Based on these fixed semantic boundaries, the model then iteratively generates the intermediate content within the block. The fixed anchors serve as guidance, ensuring clear separation between segments and thereby promoting coherent content generation.

We process the blocks sequentially through two stages. Specifically, once the length of the thinking block is determined, its content is generated; the additional information obtained from this reasoning step is then used to determine the length of the answering block, which is subsequently generated in an iterative manner. This design maximizes the benefit of the reasoning process by leveraging the information gained in the first stage to enhance the quality of the final answer. Further implementation details are provided in Algorithm 1.

Algorithm 1 DIA

Require: Input sequence $X = \{Q, X_L\}$, begin-anchor set $\mathcal{B} = \{b_1, \dots, b_{|\mathcal{B}|}\}$, end-anchor set $\mathcal{E} = \{e_1, \dots, e_{|\mathcal{E}|}\}$, confidence threshold c , expand size Δ , max length M
Ensure: Completed sequence $X = \{Q, C_1, \dots, C_{|\mathcal{B}|}\}$

- 1: Divide X_L into $|\mathcal{B}|$ blocks $\mathcal{C} = \{C_1, \dots, C_{|\mathcal{B}|}\}$, each of max length M
- 2: **for** $i \leftarrow 1$ to $|\mathcal{B}|$ **do**
- 3: Insert begin anchor b_i at the head of block C_i
- 4: **end for**

Stage 1: Generation length adjustment by estimating position of the end anchor

- 5: **for** each block C_i **do**
- 6: **while** True **do**
- 7: $Y \leftarrow \text{Infer}(Q, C_1 \dots C_i)$ *perform one diffusion-based inference*
- 8: Scan tokens of C_i from head
- 9: **if** a subsequence $y \subseteq Y$ matches some part of $e_i \in \mathcal{E}$ with $\text{Conf}(y) > c$ **then**
- 10: Truncate C_i at this position
- 11: **break**
- 12: **else if** no partial match found and $|C_i| + \Delta \leq M$ **then**
- 13: Expand C_i by Δ tokens *controlled by expand size*
- 14: **else if** $|C_i| + \Delta > M$ **then**
- 15: Stop expansion for C_i
- 16: **break**
- 17: **end if**
- 18: **end while**
- 19: Infill selected $e_i \in \mathcal{E}$ at the tail of C_i

Stage 2: Iterative Denoising with Infilling

- 20: Generate all remaining masked positions in C_i using $\text{Infer}(Q, C_1 \dots C_i)$
- 21: **end for**
- 22: **return** $X = \{Q, C_1, \dots, C_{|\mathcal{B}|}\}$

270 4 EXPERIMENTS
271272 4.1 BENCHMARKS
273274 To systematically evaluate the effectiveness of our method, we adopt two reasoning-sensitive math-
275 ematical benchmarks: **GSM8K** 0-shot and **MATH** 0-shot. **GSM8K**(Cobbe et al., 2021) is a widely
276 used dataset of grade-school math word problems, covering basic arithmetic and commonsense rea-
277 soning tasks, and thus serves as a reliable measure of a model’s performance in everyday numerical
278 reasoning scenarios. In contrast, **MATH**(Hendrycks et al., 2021) is a more challenging benchmark
279 that spans competition-level problems from elementary to advanced mathematics, encompassing
280 diverse problem types and difficulty levels, thereby providing a rigorous assessment of a model’s
281 capabilities in complex reasoning and knowledge generalization.
282

283 4.2 BASELINES

284 We select Dream-7B-Base-v0 and Dream-7B-Instruct-v0 as our baseline models. The Dream-7B
285 series is initialized from the Qwen model family and has achieved superior performance compared
286 to other open-source diffusion models on multiple benchmark tasks. To ensure fairness, all exper-
287 iments are conducted with corresponding modifications to the official codebase, without applying
288 any additional acceleration or optimization techniques.
289290 4.3 IMPLEMENTATION DETAILS
291292 Our method is implemented within the PyTorch framework. For a fair comparison, all models are
293 evaluated under the same GPU configuration when tested on identical tasks. Additional implemen-
294 tation details are provided in Appendix B.
295

296 4.4 MAIN RESULTS

297 We conduct a comprehensive evaluation on the two benchmarks. Table 1 reports the comparison
298 between our method and the baselines. Specifically, Dream-7B-Base-v0 and Dream-7B-Instruct-
299 v0 generate responses by relying solely on additional format-constrained prompts. In contrast, the
300 infilling approach inserts the corresponding anchors at designated positions within the response
301 sequence of Dream-7B-Base-v0, thereby guiding the model to produce answers.
302303 We introduce two metrics, Format Score S_{format} and Accuracy $Acc.$, for evaluation. Accuracy mea-
304 sures whether the generated response is correct, while Format Score assesses whether the response
305 adheres to the predefined format requirements.
306307 Table 1: Comparison of Methods on Format Adherence and Benchmark Performance. DIA achieves
308 the highest format scores across both **GSM8K** and **MATH**, substantially outperforming baseline and
309 infilling approaches. These results highlight the robustness and effectiveness of DIA in enforcing
310 strict structural constraints while maintaining competitive answer accuracy.
311

	0-shot GSM8K		0-shot MATH-500	
	S_{format}	Acc.	S_{format}	Acc.
Dream-7B-Base (Ye et al., 2025)	0	68.99	0	25.14
Dream-7B-Instruct (Ye et al., 2025)	0	15.01	0	25.28
Infilling Baseline	58.83	14.86	29.10	19.52
Dynamic Infilling Anchor (Ours)	72.63	46.78	76.82	20.08

312 Compared to the performance degradation introduced by the infilling approach, DIA achieves su-
313 perior results in both format adherence and answer quality. On **GSM8K**, DIA not only raises the for-
314 mat score from 58.83% to 72.63% but also substantially improves accuracy from 14.86% to 46.78%,
315 highlighting its ability to simultaneously enforce structural fidelity and enhance reasoning correct-
316 ness. On the more challenging **MATH** benchmark, DIA boosts the format score from 29.10% to
317 76.82%, demonstrating remarkable robustness in preserving structural anchors even under complex
318

problem settings, while maintaining comparable answer accuracy to baseline methods. The results clearly demonstrate that DIA addresses the shortcomings of baseline models and methods under format-constrained tasks, ensuring accurate preservation of the required format. Moreover, unlike the infilling approach, DIA’s flexible design of generation length allows each stage to maintain high answer quality, thereby achieving a better balance between performance and format correctness across diverse benchmarks.

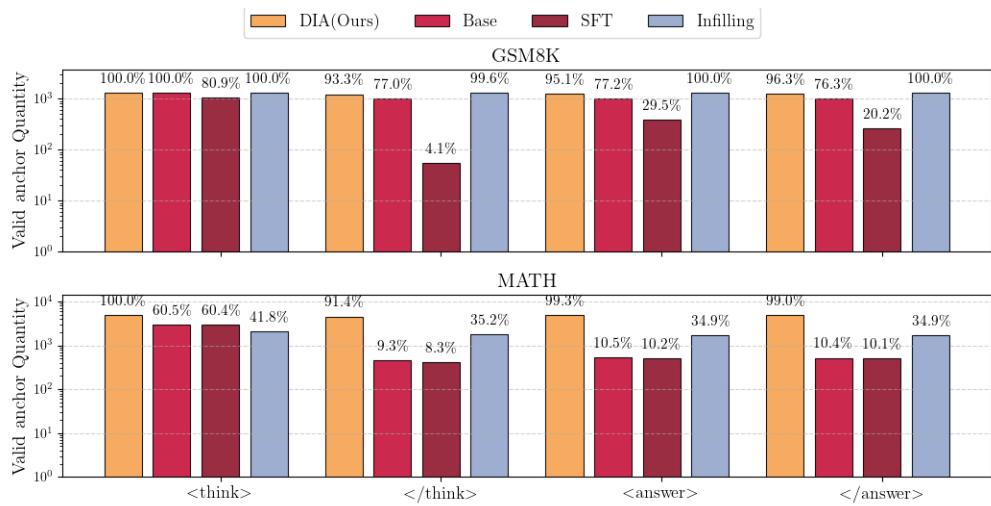


Figure 2: DIA delivers reliable anchor preservation and stable performance across different benchmarks. Even as task complexity increases on the more challenging MATH, DIA consistently maintains high anchor retention, underscoring its robustness under stricter reasoning and formatting requirements.

Figure 2 presents a detailed comparison of anchor retention ratios across different methods. Overall, DIA demonstrates outstanding stability on both GSM8K and MATH, consistently achieving nearly 100% retention across all four anchors, including both begin anchor (*<think>*) and *<answer>*) and end anchor (*</think>* and *</answer>*). This robust performance shows that the proposed two-stage generation strategy not only preserves anchors under varying conditions but also enforces strict compliance with the predefined format throughout the entire sequence. Such stability is particularly important in reasoning-oriented tasks, where structural deviations can lead to incomplete, unparseable, or misleading outputs.

In contrast, the Base and SFT models suffer from significant structural degradation. For example, on GSM8K, their retention rates for *</think>* collapse to only 4.4% and 29.5%, respectively, and on MATH, the rates for *</think>* and *</answer>* drop to single digits. These results reveal a consistent failure of conventional methods to maintain boundary integrity, especially in longer or more complex reasoning chains, where models tend to lose track of global structure and generate unbalanced outputs. Such issues undermine the reliability of the generated content and illustrate why relying solely on prompt-based constraints or fine-tuning strategies is insufficient for strict format adherence.

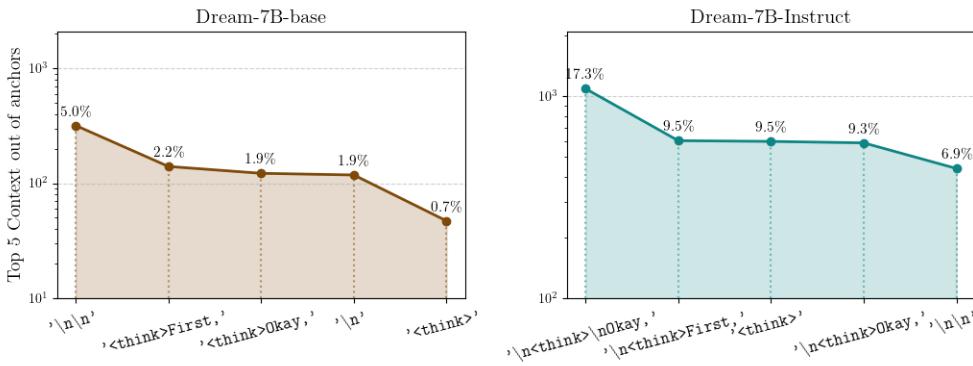
Although the Infilling baseline achieves higher anchor retention than Base and SFT—nearly matching DIA on GSM8K for *<think>* and *<answer>*, its performance on begin anchors remains unstable. Crucially, this preservation does not translate into gains in overall format correctness or answer accuracy. For instance, while Infilling retains anchors on GSM8K, its downstream results remain far below DIA in both structural and semantic evaluations. This mismatch highlights that simply inserting anchors is not enough; without a dynamic mechanism for allocating and regulating generation space, models either over-generate redundant tokens or fail to stop at the correct boundaries.

Taken together, these results provide a fine-grained validation of Table 1. They show that DIA not only outperforms existing approaches in aggregate metrics but also secures overwhelming superiority in preserving critical anchors across diverse datasets. By ensuring that every anchor is faithfully

378 retained, DIA substantially enhances the reliability of format-constrained generation, laying a foun-
 379 dation for robust application of dLLMs in reasoning, structured reporting, and other scenarios where
 380 strict adherence to format is essential.

382 4.5 ANALYSIS EXPERIMENTS

384 4.5.1 BEHAVIOR OUTSIDE ANCHOR CONTEXTS



399 Figure 3: Top-5 statistics of out-of-anchor content generated by baseline models across different
 400 benchmarks. The baseline models fail to establish effective semantic boundaries aligned with anchor
 401 positions, leading to unconstrained content generation.

402
 403 To more comprehensively evaluate model performance under format-constrained tasks, we con-
 404 ducted a statistical analysis of the responses generated by Dream-7B-Base-v0 and Dream-7B-
 405 Instruct-v0 on two benchmarks. Our analysis focuses on the content appearing beyond the
 406 $</\text{answer}>$ anchor boundary, as this indicates whether the models can effectively leverage the
 407 semantic boundaries established by anchors to properly constrain their generation. Specifically, we
 408 examined all responses across the two benchmarks and extracted the Top-5 most frequent continu-
 409 ations occurring after the $</\text{answer}>$ boundary for both models, with the results shown in Figure
 410 3. This analysis provides a fine-grained perspective on boundary robustness, complementing aggre-
 411 gate metrics by revealing how models behave when the intended termination point has already been
 412 reached.

413 As shown in Figure 3, Dream-7B-Base-v0 produces dispersed and low-frequency redundancy be-
 414 yond the $</\text{answer}>$ anchor, with all Top-5 patterns below 6%, whereas Dream-7B-Instruct-v0
 415 exhibits more concentrated redundancy, with Top-5 patterns reaching up to 17.3% and dominated by
 416 repeated $<\text{think}>$ tokens. The contrast highlights that the Base model tends toward uncontrolled
 417 drifting, while the Instruct variant systematically re-enters the reasoning phase, reflecting a struc-
 418 tural weakness in anchor boundary enforcement. Overall, the Base model lacks effective boundary
 419 control, while the Instruct model suffers from patterned continuations, and both fail to reliably ter-
 420 minate at the anchor—underscoring the necessity of DIA in eliminating out-of-anchor redundancy
 421 and ensuring format adherence. Importantly, such failures not only compromise readability but also
 422 propagate errors to downstream applications that rely on strictly bounded outputs.

423 4.5.2 EXPAND TIMES

424 To establish a reasonable upper bound for the maximum block length, we analyzed the number of
 425 extensions in the reasoning part of all format-correct responses. The details are presented in Figure
 426 4. The results show that the chosen maximum length threshold effectively ensures the allocation of
 427 appropriate generation space. Specifically, the observed extension counts fall within the range of
 428 (30, 85), which is substantially smaller than the number of extensions permitted by the maximum
 429 threshold. In other words, although a large upper bound is allowed, the vast majority of responses
 430 naturally converge to a much smaller range of expansions, confirming that the setting of the maxi-
 431 mum block length is both sufficient and not overly restrictive.

432 Moreover, the proportion of format-correct re-
 433 sponses within this range consistently exceeds
 434 90%, further validating both the effectiveness
 435 of our method and the appropriateness of the
 436 current threshold setting. Importantly, this
 437 trend is observed across both GSM8K and
 438 MATH, where over 94% of samples fall into
 439 the effective expansion range, indicating that
 440 DIA adapts reliably to tasks of different scales
 441 and difficulties. Such stability suggests that
 442 the block-length constraint not only prevents
 443 degenerate over-expansion but also preserves
 444 high-quality structural adherence across bench-
 445 marks.

5 DISCUSSIONS

446 While Dynamic Infilling Anchors (DIA) have
 447 shown strong effectiveness in improving format-constrained generation, several limitations remain.
 448 First, the current method relies on manually specified anchors, assuming that task boundaries and
 449 semantic roles (*e.g.*, reasoning vs. answering) are fixed. In more complex tasks such as open-domain
 450 dialogue or multi-stage reasoning, anchors may not follow stable positions, and their semantics may
 451 shift with context. Extending DIA to automatically infer or adapt anchor definitions remains an
 452 open challenge. Second, DIA introduces inference overhead from iterative space adjustments. Al-
 453 though modest in our experiments, scaling to longer outputs or interactive systems may require more
 454 efficient mechanisms for anchor prediction and length control. Finally, our evaluation is limited to
 455 reasoning datasets like GSM8K and MATH; whether DIA generalizes to long-form writing, program
 456 synthesis, or multimodal tasks is still uncertain.

457 Despite these limitations, the anchor-based perspective also suggests new opportunities. Anchors
 458 need not be restricted to reasoning and answering boundaries; they could represent higher-level
 459 structures such as proof steps, code blocks, or multimodal transitions. In tasks where anchor seman-
 460 tics evolve dynamically, DIA could be extended with adaptive or hierarchical anchor systems that
 461 refine themselves during generation. Combining DIA with lightweight training or alignment meth-
 462 ods may also help models acquire richer anchor semantics, reducing manual specification while
 463 improving robustness across domains. These possibilities suggest that anchors are not only a con-
 464 trol mechanism for current benchmarks but also a lens for rethinking how diffusion language models
 465 manage structure and meaning in more complex generative scenarios.

6 CONCLUSION

466 In this work, we introduced Dynamic Infilling Anchors (DIA), a training-free method for enhancing
 467 format-constrained generation in diffusion language models (dLLMs). By adopting a two-stage
 468 decoding strategy—length expansion guided by end-anchor prediction, followed by explicit anchor
 469 completion and content generation—DIA achieves a strong balance between structural fidelity and
 470 semantic quality. Experiments on reasoning benchmarks such as GSM8K and MATH show that DIA
 471 substantially improves format adherence while maintaining competitive answer accuracy, validating
 472 its effectiveness in tasks requiring both precision and reliability.

473 Beyond these empirical results, our study underscores the broader potential of dLLMs for structured
 474 text generation without additional training. Leveraging their intrinsic awareness of generation space,
 475 DIA demonstrates that such models can be steered to meet strict output constraints, opening oppor-
 476 tunities in code generation, structured proofs, and machine-readable reporting. At the same time,
 477 DIA points to several avenues for future research, including automated anchor design, extensions to
 478 hierarchical or nested constraints, and integration with lightweight training or alignment techniques.
 479 By bridging structural control and semantic quality, DIA lays the groundwork for deploying dLLMs
 480 as dependable reasoning assistants and as general-purpose models in real-world applications where
 481 consistency, interpretability, and reliability are paramount.

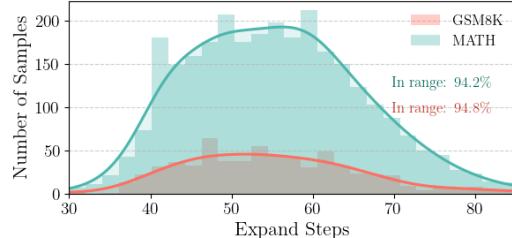


Figure 4: Statistics of effectively expanded samples. The maximum length threshold ensures that the vast majority of cases receive an appropriate number of expansions, thereby safeguarding answer quality.

486 ETHICS STATEMENT
487488
489 This research does not involve human participants, personally identifiable information, or sensitive
490 user data. All experiments are carried out on publicly available datasets, namely GSM8K and
491 MATH, which are widely adopted benchmarks for assessing the reasoning ability of large language
492 models. These datasets consist of synthetic or anonymized problem–solution pairs and do not con-
493 tain private or proprietary information, thereby avoiding risks associated with data collection or
494 misuse.495 As with any work involving large language models, we acknowledge potential ethical concerns. A
496 risk of misuse remains: models augmented with improved format-adherence mechanisms could be
497 applied in contexts where reasoning outputs are consumed without adequate verification, potentially
498 leading to the dissemination of incorrect or misleading content. We emphasize that our proposed
499 method should only be deployed in scenarios where outputs are subject to human oversight or rigor-
500 ous automated validation. While our method focuses on structural control rather than altering model
501 parameters, it inherits the biases and limitations of the underlying pretrained models. Although Dy-
502 namic Infilling Anchors (DIA) does not amplify such issues, users of this technique should remain
503 mindful of the broader ethical challenges associated with large-scale language models.504 We stress the importance of responsible deployment. The intention of this work is to advance re-
505 search on reliable, format-constrained generation and to provide the community with a training-free
506 mechanism that improves structural fidelity without additional fine-tuning or data collection. We do
507 not endorse its use in high-stakes domains such as medical diagnosis, legal reasoning, or automated
508 decision-making without extensive domain-specific evaluation and safeguards. By situating our con-
509 tributions within established ethical guidelines in AI research, we aim to ensure that the benefits of
510 this work are realized while minimizing risks of harm.511
512 REPRODUCIBILITY STATEMENT
513514
515 We have taken comprehensive steps to maximize the reproducibility of our results and to facil-
516 tiate independent verification by other researchers. To this end, we adhere to several principles of
517 transparent and replicable research practice.518
519 The appendix provides extensive details about our experimental setup, including hardware config-
520 uration (GPU model, number of devices, and memory constraints), software environment (Python
521 and PyTorch versions, dependencies, and library compatibility), parameter choices (generation hy-
522 perparameters, block sizes, anchor thresholds, and sampling strategies), and evaluation protocols
523 (dataset splits, metrics, and scoring procedures). These details are presented to eliminate ambiguity
524 and enable replication.525
526 The implementation of Dynamic Infilling Anchors (DIA) is based on the official Dream-7B code-
527 base with minimal modifications, all of which are clearly documented. We describe both algorithmic
528 adjustments, such as the two-stage anchor-based procedure for length expansion and truncation, and
529 essential engineering decisions including input preprocessing and memory optimization strategies.
530 Such documentation lowers the barrier for others to replicate our findings without requiring exten-
531 sive reverse engineering.532
533 We also commit to releasing all necessary artifacts for reproduction. This includes the source code,
534 scripts for data preprocessing and evaluation, and configuration files for running experiments. The
535 release will be accompanied by instructions for environment setup and guidelines for reproducing
536 results on GSM8K and MATH. Seed values and randomization settings will be provided to ensure
537 consistent outputs across trials.538
539 In line with community standards, long-term accessibility is emphasized. All released materials
540 will be archived in a permanent public repository with version control, ensuring that they remain
541 accessible as dependencies evolve. By combining detailed documentation, transparent reporting,
542 and open resources, we aim to make our work fully reproducible and to encourage further extensions
543 and critical evaluation by the research community.

540 REFERENCES
541

542 Anthropic. Introducing claude 4, 2025. URL <https://www.anthropic.com/news/claude-4>.

543

544 Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models, 2025. URL <https://arxiv.org/abs/2503.09573>.

545

546

547 Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2023. URL <https://arxiv.org/abs/2107.03006>.

548

549

550 Debangshu Banerjee, Tarun Suresh, Shubham Ugare, Sasa Misailovic, and Gagandeep Singh. Crane: Reasoning with constrained llm generation, 2025. URL <https://arxiv.org/abs/2502.09061>.

551

552

553

554 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.

555

556

557

558 Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model, 2024. URL <https://arxiv.org/abs/2306.16092>.

559

560

561

562 Google Deepmind. Gemini diffusion: Our state-of-the-art, experimental text diffusion model, 2024.

563 URL <https://deepmind.google/models/gemini-diffusion/>.

564

565 Google Deepmind. Gemini 2.5: Our most intelligent ai model, 2025.

566 URL <https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/>.

567

568 DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanja Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL <https://arxiv.org/abs/2412.19437>.

594 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep
 595 bidirectional transformers for language understanding, 2019. URL <https://arxiv.org/abs/1810.04805>.
 596

597

598 Jingtong Gao, Bo Chen, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo,
 599 Ruiming Tang, and Xiangyu Zhao. Llm4rerank: Llm-based auto-reranking framework for rec-
 600 ommendations, 2025. URL <https://arxiv.org/abs/2406.12433>.
 601

602 Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An,
 603 Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language
 604 models via adaptation from autoregressive models, 2025a. URL <https://arxiv.org/abs/2410.17891>.
 605

606 Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and
 607 Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code gen-
 608 eration, 2025b. URL <https://arxiv.org/abs/2506.20639>.
 609

610 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad
 611 Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan,
 612 Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Ko-
 613 rennev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava
 614 Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux,
 615 Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret,
 616 Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius,
 617 Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary,
 618 Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab
 619 AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco
 620 Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind That-
 621 tai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Kore-
 622 vaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra,
 623 Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Ma-
 624 hadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Janya Lee, Jeremy Fu,
 625 Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jong-
 626 soo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala,
 627 Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid
 628 El-Arini, Kritika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhota, Lauren
 629 Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin,
 630 Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi,
 631 Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew
 632 Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Ku-
 633 mar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoy-
 634 chev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan
 635 Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan,
 636 Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ra-
 637 mon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Ro-
 638 hit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan
 639 Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell,
 640 Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng
 641 Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer
 642 Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman,
 643 Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mi-
 644 haylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor
 645 Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei
 646 Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang
 647 Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Gold-
 schlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning
 Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh,
 Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajinfeld, Adithya Gangidi, Adolfo Victoria,
 Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenber, Alexei Baevski, Allie Feinstein,

648 Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, An-
 649 drew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, An-
 650 nie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel,
 651 Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leon-
 652 hardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu
 653 Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Mon-
 654 talvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao
 655 Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia
 656 Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide
 657 Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le,
 658 Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily
 659 Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smo-
 660 thers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni,
 661 Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia
 662 Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan,
 663 Hakan Inan, Hamid Shojaanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harri-
 664 son Rudolph, Helen Suk, Henry Aspegen, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj,
 665 Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James
 666 Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jen-
 667 nifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang,
 668 Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Jun-
 669 jie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy
 670 Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang,
 671 Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell,
 672 Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa,
 673 Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias
 674 Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L.
 675 Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike
 676 Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari,
 677 Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan
 678 Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong,
 679 Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent,
 680 Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar,
 681 Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Ro-
 682 driguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy,
 683 Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin
 684 Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon,
 685 Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ra-
 686 maswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha,
 687 Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal,
 688 Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satter-
 689 field, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj
 690 Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo
 691 Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook
 692 Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Ku-
 693 mar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihalescu, Vladimir Ivanov,
 694 Wei Li, Wencheng Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiao-
 695 jian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia,
 696 Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao,
 697 Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhao-
 698 duo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL
 699 <https://arxiv.org/abs/2407.21783>.

697 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
 698 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.
 699 URL <https://arxiv.org/abs/2103.03874>.

700 Jaehyeong Jo and Sung Ju Hwang. Continuous diffusion model for language modeling, 2025. URL
 701 <https://arxiv.org/abs/2502.11564>.

702 Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child,
 703 Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language
 704 models, 2020. URL <https://arxiv.org/abs/2001.08361>.

705

706 Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer
 707 Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and
 708 Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion, 2025. URL
 709 <https://arxiv.org/abs/2506.17298>.

710

711 Jinsong Li, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jiaqi Wang, and Dahu Lin. Beyond fixed:
 712 Training-free variable-length denoising for diffusion large language models, 2025. URL <https://arxiv.org/abs/2508.00819>.

713

714 Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-
 715 training with frozen image encoders and large language models, 2023. URL <https://arxiv.org/abs/2301.12597>.

716

717 Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL
 718 <https://arxiv.org/abs/2304.08485>.

719

720 Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke
 721 Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?,
 722 2022. URL <https://arxiv.org/abs/2202.12837>.

723

724 Niels Mündler, Jingxuan He, Hao Wang, Koushik Sen, Dawn Song, and Martin Vechev. Type-
 725 constrained code generation with language models. *Proceedings of the ACM on Programming
 726 Languages*, 9(PLDI):601–626, June 2025. ISSN 2475-1421. doi: 10.1145/3729274. URL <http://dx.doi.org/10.1145/3729274>.

727

728 Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin,
 729 Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025. URL <https://arxiv.org/abs/2502.09992>.

730

731 OpenAI. Introducing gpt-5, 2025. URL <https://openai.com/index/introducing-gpt-5/>.

732

733 Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong
 734 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kel-
 735 ton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike,
 736 and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.
 737 URL <https://arxiv.org/abs/2203.02155>.

738

739 Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and
 740 Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model,
 741 2024. URL <https://arxiv.org/abs/2305.18290>.

742

743 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy
 744 optimization algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.

745

746 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
 747 Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathe-
 748 matical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.

749

750 Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang
 751 Yang, Hongli Yu, Xingwei Qu, Yuwei Fu, Jing Su, Ge Zhang, Wenhao Huang, Mingxuan Wang,
 752 Lin Yan, Xiaoying Jia, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Yonghui Wu, and Hao Zhou.
 753 Seed diffusion: A large-scale diffusion language model with high-speed inference, 2025. URL
 754 <https://arxiv.org/abs/2508.02193>.

755

Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing re-
 756 infornce learning framework for diffusion large language models, 2025. URL <https://arxiv.org/abs/2509.06949>.

756 xAI. Grok 4, 2025. URL <https://x.ai/news/grok-4>.
 757

758 Honglin Xiong, Sheng Wang, Yitao Zhu, Zihao Zhao, Yuxiao Liu, Linlin Huang, Qian Wang, and
 759 Dinggang Shen. Doctorglm: Fine-tuning your chinese doctor is not a herculean task, 2023. URL
 760 <https://arxiv.org/abs/2304.01097>.

761 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang
 762 Gao, Chengan Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu,
 763 Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin
 764 Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang,
 765 Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui
 766 Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang
 767 Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger
 768 Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan
 769 Qiu. Qwen3 technical report, 2025a. URL <https://arxiv.org/abs/2505.09388>.

770 Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large
 771 language models, 2023. URL <https://arxiv.org/abs/2306.06031>.
 772

773 Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada:
 774 Multimodal large diffusion language models, 2025b. URL <https://arxiv.org/abs/2505.15809>.
 775

776 Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng
 777 Kong. Dream 7b: Diffusion large language models, 2025. URL <https://arxiv.org/abs/2508.15487>.
 778

779 Qinyuan Ye, Maxamed Axmed, Reid Pryzant, and Fereshte Khani. Prompt engineering a prompt
 780 engineer, 2024. URL <https://arxiv.org/abs/2311.05661>.
 781

782 Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan
 783 Li. Llada-v: Large language diffusion models with visual instruction tuning, 2025. URL <https://arxiv.org/abs/2505.16933>.
 784

785 Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion
 786 large language models via reinforcement learning, 2025. URL <https://arxiv.org/abs/2504.12216>.
 787

788 Shengyao Zhuang, Xueguang Ma, Bevan Koopman, Jimmy Lin, and Guido Zuccon. Rank-r1:
 789 Enhancing reasoning in llm-based document rerankers via reinforcement learning, 2025. URL
 790 <https://arxiv.org/abs/2503.06034>.
 791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810 A NOTATION SUMMARY
811
812
813
814

Table 2: Notation Summary

815 Symbol	816 Description
817 $X = \{Q, X_L\}$	818 Input sequence consisting of query Q and all-masked sequence X_L
819 Q	820 Input query provided to the diffusion-based LLM
821 X_L	822 Fully masked sequence to be partitioned into blocks
823 $\mathcal{B} = \{b_1, \dots, b_{ \mathcal{B} }\}$	824 Set of begin anchors inserted at block heads
825 $\mathcal{C} = \{C_1, \dots, C_{ \mathcal{B} }\}$	826 Set of blocks obtained from X_L
827 $\mathcal{E} = \{e_1, \dots, e_{ \mathcal{E} }\}$	828 Set of end anchors to be infilled at block tails
829 C_i	830 The i -th block after begin anchor insertion and expansion
831 $ C_i $	832 Current length of block C_i during expansion
833 y	834 Predicted subsequence scanned within block C_i
835 c	836 Confidence threshold for partial end-anchor detection
837 Δ	838 Expand size: number of tokens added per expansion step
839 M	840 Maximum allowed length of a block C_i
841 $\text{Infer}(X)$	842 Diffusion-based inference function applied to sequence X
843 $\text{Conf}(y)$	844 Confidence score assigned to subsequence y

830
831
832 B EXPERIMENTAL SETTINGS
833
834
835

Table 3: Experimental Settings

836 Category	837 Setting
838 Model	839 Dream-7B-Base / Dream-7B-Instruct
840 Datasets(Max new tokens)	841 GSM8K(256), MATH-500(512)
841 Evaluation Metrics	842 Format Score, Answer Score, Anchor Retention Rate
842 Confidence Threshold c	843 0.065,0.05
843 Expand Size Δ	844 4,4
844 Max Block Length M	845 512,512
845 Diffusion Steps	846 512,512
846 Batch Size	847 1,3
847 Total Samples	848 GSM8K: 1319, MATH-500: 5000
848 Hardware	849 NVIDIA vGPU-32G, vGPU-48G
849 Framework	850 PyTorch 2.5.1, Python 3.10.16

851
852 C CASE STUDY
853

854 C.1 BASE MODEL

855 "query": "Brandon's iPhone is four times as old as Ben's iPhone.
 856 Ben's iPhone is two times older than Suzy's iPhone. If Suzy's
 857 iPhone is 1 year old, how old is Brandon's iPhone?", "method":
 858 "Base", "output": "<think> Suzy's iPhone is 1 year old. Ben's
 859 iPhone is two times older than Suzy's iPhone, so Ben's iPhone is 2
 860 years old. Brandon's iPhone is four times as old as Ben's iPhone,
 861 so Brandon's iPhone is 4 years old. So the answer is 4 years
 862 old.</think><answer> 4 </answer>\n\n", "flaw": "Content out of
 863 anchor"

```

864 "query": "Well's mother sells watermelons, peppers, and oranges
865 at the local store. A watermelon costs three times what each
866 pepper costs. An orange costs 5 less than what a watermelon cost.
867 Dillon is sent to the store to buy 4 watermelons, 20 peppers, and
868 10 oranges. What's the total amount of money he will spend if
869 each pepper costs 15$?", "method": "Base", "output":
870 "<<think>First, I need to find the cost of a watermelon. Since a
871 watermelon costs three times what each pepper costs, and each
872 pepper costs 15$, I can multiply 15 by 3 to get the cost of a
873 watermelon.  $15 * 3 = 45$ . So a watermelon costs 45$.\n\nNext, I
874 need to find the cost of an orange. Since an orange costs 5 less
875 than what a watermelon costs, and a watermelon costs 45$, I can
876 subtract 5 from 45 to get the cost of an orange.  $45 - 5 = 40$ . So
877 an orange costs 40$.\n\nNow, I need to find the total amount of
878 money Dillon will spend. He is buying 4 watermelons, 20 peppers,
879 and 10 oranges. I can calculate the total cost by multiplying the
880 cost of each item by the number of items and adding them
881 together.\n\nFor the watermelons:  $4 * 45 = 180$ \nFor the peppers:
882  $20 * 15 = 300$ \nFor the oranges:  $10 * 40 = 400$ ", "flaw": "Missing
883 anchors"
```

C.2 INSTRUCT MODEL

```

884 "query": "Peter plans to go to the movies this week. He always
885 gets a ticket for $7 and popcorn for $7. If he has 42 dollars for
886 the week, how many times can he go to the movies?", "method":
887 "Instruct", "output": "<think><answer>3 times</answer>", "flaw":
888 "Missing anchors. Incomplete response."
```

```

889 "query": "Peter plans to go to the movies this week. He always
890 gets a ticket for $7 and popcorn for $7. If he has 42 dollars for
891 the week, how many times can he go to the movies?", "method":
892 "Instruct", "output": "" "$21</think>", "flaw": "Missing
893 anchors. Incomplete response."
```

C.3 INFILLING METHOD

```

894 "query": "Peter plans to go to the movies this week. He always
895 gets a ticket for $7 and popcorn for $7. If he has 42 dollars for
896 the week, how many times can he go to the movies?", "method":
897 "Infilling method", "output": "" <think>...</think>
898 <answer>....</answer> <think>.</think> <answer>....</answer>
899 <think>....</think> <answer>....</answer> <think>....</think>
900 <answer>....</answer> <think>....</think> <answer>....</answer>
901 <think>....</think> <answer>....</answer> <think>....</think>
902 <answer>....</answer> <think>....</think> <answer>....</answer>
903 <think>....</think> <answer>....</answer> <think>....</think>
904 <answer>....</answer> <think>....</think> <answer>....</answer>
905 <think>....</think> <answer>....</answer>
906 <think>....</think> <answer>....</answer> <think>....</think>
907 <answer>....</answer> <think>....</think> <answer>....</answer>
908 <think>....</think> <answer>....</answer> <think>....</think>
909 <answer>....</answer> <think>....</think> <answer>....</answer>
910 <think>....</think> <answer>....</answer>
911 <think>....</think> <answer>....</answer> "", "flaw": "Broken
912 response."
```

```

913 "query": "Kyle bought last year's best-selling book for $19.50.
914 This is with a 25% discount from the original price. What was the
915 original price of the book?", "method": "Infilling method",
916 "output": "" <think>25</think> <answer>25</answer> \nThe
917 question:\n", "flaw": "Content out of anchor. Incomplete
918 response."
```

918
919

C.4 DIA

920 "query": "Charlie wants to sell beeswax candles. For every pound
 921 of beeswax, he can make 10 tapered candles. One pound of beeswax
 922 and the wicks cost \$10.00 in supplies. If he sells each candle
 923 for \$2.00 each, what is his net profit if he makes and sells 20
 924 candles?", "method": "DIA", "output": ""<think>First, I need to
 925 figure out how much candles Charlie can make with 20 candles.
 926 Since he can make 10 candles from one pound of beeswax, he can
 927 make 20 candles from 2 pounds of beeswax. The pound of beeswax
 928 costs \$10, so 2 pounds will cost \$20. The total of supplies is
 929 \$20. Now, I need to calculate out how much money he will make
 930 from selling 20 candles. Since he sells each candle for \$2, he
 931 will make $20 \times \$2 = \40 . Now, I need to subtract the cost of
 932 supplies from the amount he will make from selling the candles.
 933 So, $\$40 - \$20 = \$20$. Therefore, Charlie will profit a net of
 934 \$20.00 and makes and sells 20 candles0</think><answer>
 20</answer> "", "flaw": "None."

935 "query": "John drives for 3 hours at a speed of 60 mph and then
 936 turns around because he realizes he forgot something very
 937 important at home. He tries to get home in 4 hours but spends the
 938 first 2 hours in standstill traffic. He spends the next half-hour
 939 driving at a speed of 30mph, before being able to drive the
 940 remaining time of the 4 hours going at 80 mph. How far is he from
 941 home at the end of those 4 hours?", "method": "DIA", "output":
 942 ""<think>First, I need to calculate the distance John traveled
 943 away. He drove for 3 hours at 60 mph, so the distance is $60 \times 3 =$
 944 180 miles. Then, he turned around and tried to get home in 4
 945 hours. He, spent the first 2 hours in standstill traffic, so he
 946 didn't cover any distance during that time. He, then spent the
 947 next half-hour driving at 30 mph, so the covered distance is $30 \times$
 948 0.5 = 15 miles. He, spent the remaining 2 hours driving at 80
 949 mph, so the covered distance is $80 \times 2 = 160$ miles. The, the
 950 total distance he covered while coming back is $15 + 160 = 175$
 951 miles. Since, he traveled 180 miles away from home and then
 952 covered 175 miles back,,, he is distance from home at $180 - 175 =$
 953 5 miles. the end of those 4 hours, he, he, 5555555555555555
 miles5</think><answer>5 miles</answer> "", "flaw": "**Generation**
 954 **length prediction not completely accurate.**"

955
956

D DLLM PROMPT TEMPLATE

957
958

Table 4: Example of Prompt Design

959

Field	Content
Instruction	You are a helpful assistant that helps the user to solve the question.
Output Format	You need to think first and then answer the question briefly by following the format:<think>...</think><answer>...</answer>.
Input	Here are the questions: {QUESTION}

965
966

967

E USAGE OF LLMs

968

969
970
971

In accordance with the ICLR guidelines, we disclose the use of large language models (LLMs) in the preparation of this paper. LLMs were employed exclusively as a writing assistance tool for language polishing, grammar refinement, and improving readability. They were not involved in research ideation, experimental design, data analysis. All technical ideas, theoretical developments,

972 proofs, and experimental results presented in this paper are entirely the work of the authors. The
973 authors take full responsibility for the accuracy and integrity of the final submission.
974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025